

FIG. 2

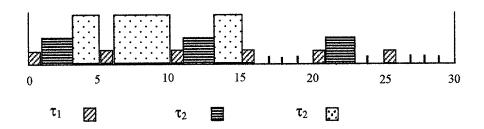


FIG. 3

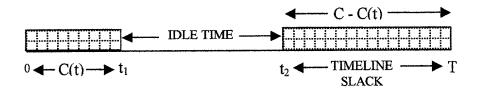


FIG. 4

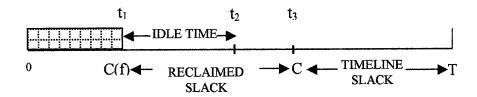
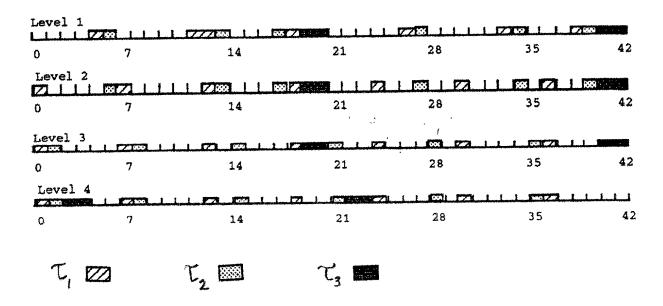
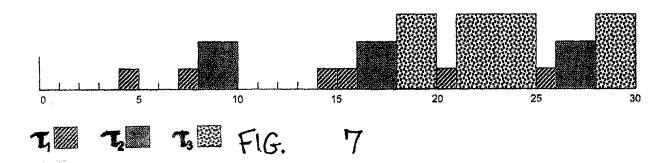
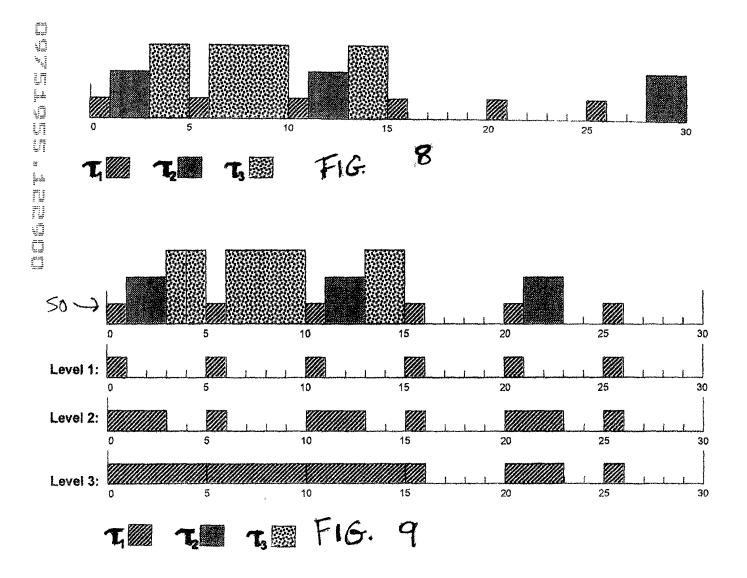


FIG. 5



F16. 6





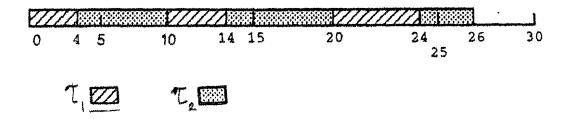
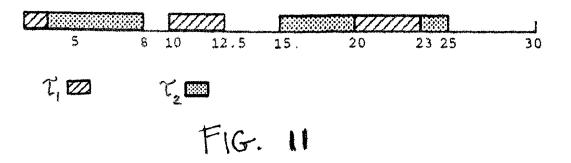
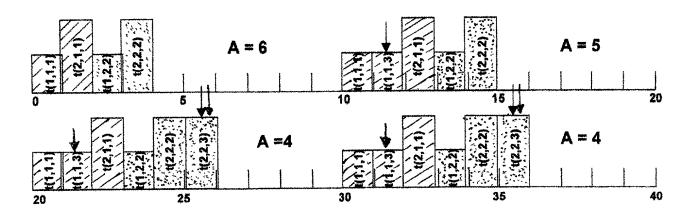


FIG. 10

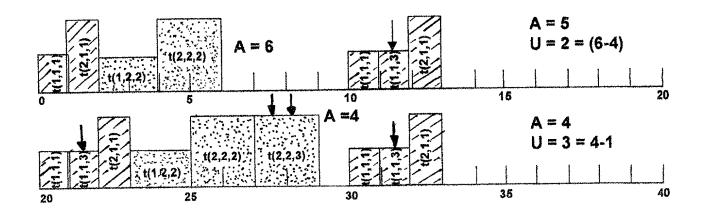




RIZZ

R2 2

FIG. 12

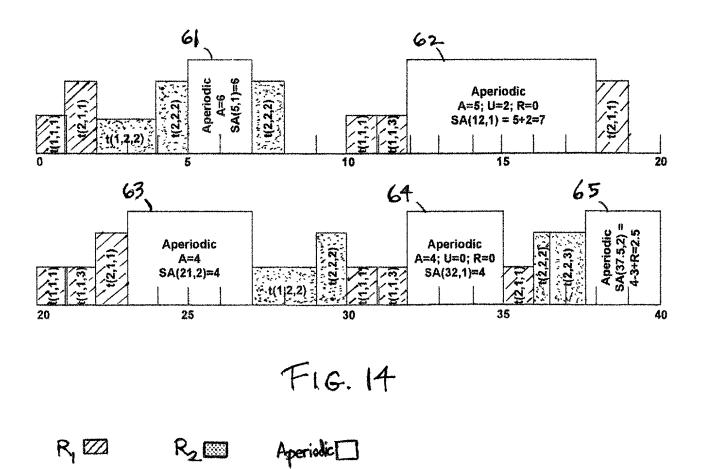


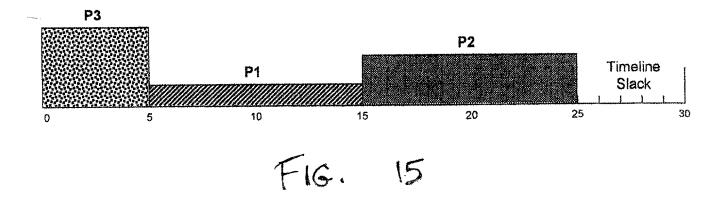
 $R_i \square$

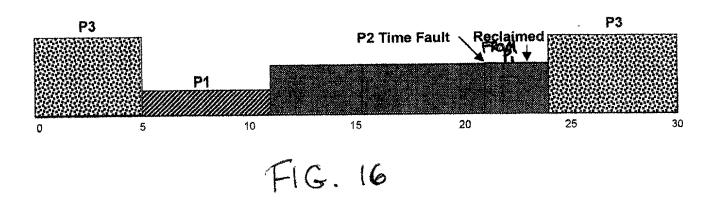
R.

FIG. 13

R,







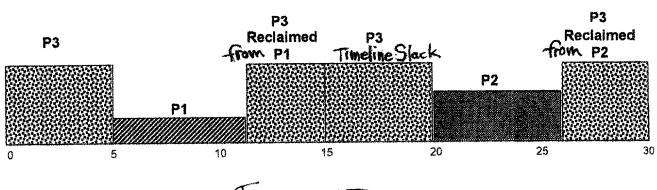
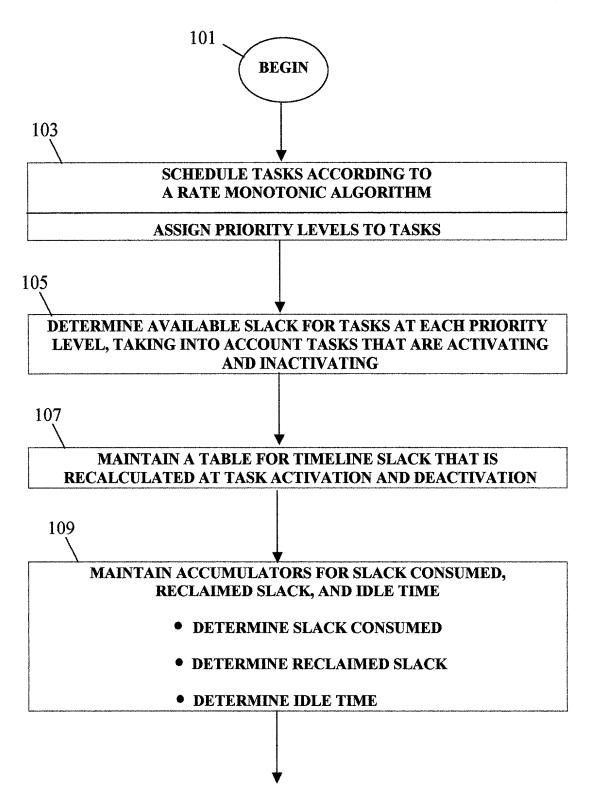


FIG. 17



UPDATE THE ACCUMULATORS UPON THE OCCURRENCE OF AN

• WHEN CROSSING A PERIOD BOUNDARY

EVENT FROM THE FOLLOWING GROUP:

- WHEN A TASK COMPLETES FOR PERIOD WHEN EXECUTING ON A FIXED BUDGET WITH SLACK TO BE RECLAIMED
- WHEN A PROCESSOR EXECUTING THE TASKS TRANSITIONS FROM IDLE TO BUSY
- WHEN A TASK COMPLETES FOR PERIOD WHEN EXECUTING ON SLACK
- PRIOR TO CALCULATING AVAILABLE SLACK FOR A NEW SLACK-CONSUMING TASK
- BEFORE MUTEX EXECUTION BY A SLACK-CONSUMING THREAD

PREDECREMENT THE ACCUMULATORS TO ALLOW FOR OVERHEAD ASSOCIATED WITH ALLOCATING SLACK, SUCH AS

CONTEXT SWAPS, CACHE REFRESHES, ETC.

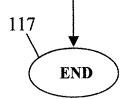
115

113

111

ALLOCATE SLACK TO TASKS IN ORDER OF PRIORITY

AN APERIODIC HIGH PRIORITY TASK CAN STEAL SLACK FROM AVAILABLE SLACK WITHOUT IMPACTING AN EXECUTION DEADLINE OF A PERIODIC LOW PRIORITY TASK



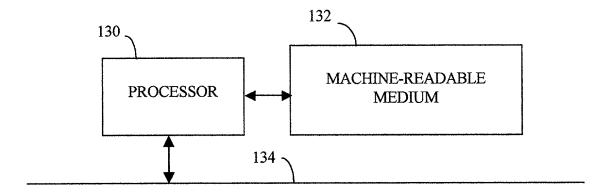


FIG. 19